NWG/RFC #102
NIC #5763

# OUTPUT OF THE HOST/HOST PROTOCOL GLITCH CLEANING COMMITTEE

22, 23 February 1971

at BBN, Cambridge

Steve Crocker
Chairman

At the NWG meeting in Urbana on 17-19 February 1971, a committee was established to look at the Host/Host protocol and see what changes were immediately desirable or necessary.

The committee is chaired by Steve Crocker, and has eight other members:

| | |
|---|---|
| Ray Tomlinson | BBN (Tenex) |
| Jim White | UCSB |
| Gary Grossman | Illinois |
| Tom Barkalow | Lincoln (TX2) |
| Will Crowther | BBN (IMPs) |
| Bob Bressler | MIT (Dynamic Modeling) |
| Doug McKay | IBM (Yorktown) |
| Dan Murphy | BBN (Tenex) |

A number of topics were discussed. On some of these topics, a consensus was reached on whether or not to recommend a change, and if so, what the change should be. On the remaining topics, specific alternatives were proposed but no consensus was reached.

The committee will immediately canvas the network community and gather reaction to its recommendations and the proposed alternatives. The committee will then reconvene at UCLA on 8 March 1971 and decide on final recommendations. Steve Crocker will then write Document #2. This sequence is in lieu of the change procedure outlined in NWG/RFC 53.

Specific Recommendations

1. The ECO and ERP command should each be 8 bits long.

2. The ERR command should be 96 bits long.

3. Message Data Types should be eliminated. Third-level protocol people may reinstate such a mechanism.

4. The Cease mechanism should be discontinued.

5. A new pair of one byte commands RST (reset) and RRP (reset reply) should be added. The RST should be interpreted as a signal to purge the NCP tables of any existing entries which arose from the sending Host. The RRP command should be returned to acknowledge receipt of the RST. The Host sending the RST may proceed after receiving either a RST or a RRP in return. A RST may be returned if the second Host comes up after the first Host.

6. Although it was suggested at the Urbana meeting that connections should be full-duplex, the committee recommends against this change.

7. Messages should be an integral number of bytes, and the number of bytes and the byte size should be specified in each message. The marking convention should be abandoned and the padding ignored.

The number of bytes in the message should be a 16-bit number following the leader. The byte size should be in the next 8-bit field. Two suggestions were generated for the starting point of the text, and these are explained in the next section.

For flow control purposes, the number of bits in a message is the product of the number of bytes and the byte size. The leader and other fixed format fields are not counted.

8. The problem of synchronizing the interrupt signal in a console input stream was considered. We consider the console input scanner as a process and note two reasonable implementations: it may either read characters as fast as it can, looking for the interrupt character and throwing away characters if there is no room in the user process' input queue; it may read characters only as fast as the user process can receive them, (or at least has room for them).

The first implementation guarantees that the interrupt character (e.g., control - C on the PDP-10 10/50) will always be acted on, but requires that the using process interpret the output stream to detect when it is sending too fast. The second implementation avoids overrun but may not allow for sending an interrupt code. Note that in the first case, allocation is always renewed as soon as possible by the console input interpreter; whereas in the second case, allocation is

renewed only as the result of acceptance of data by the user process.

We decided that this is really a third-level protocol matter, viz, use the INS to mean that a special code has been inserted into the input stream. In conjunction with this, create the special code to be put into the input sequence.

*This special code would be network-wide and independent of the particular interrupt character peculiar to the serving system.* The scheme for interrupting a serving process is that the using process inserts the serving Host's interrupt sequence, followed by the network special code, and also issue the INS.

## UNRESOLVED ALTERNATIVES

### 1. Length of Control Messages

In accordance with other specifications, control messages should be an integral number of 8-bit bytes, the length should be specified in the byte count field, and control commands should not be split across messages.

Unresolved was whether to specially limit the length of control messages. The two choices are:

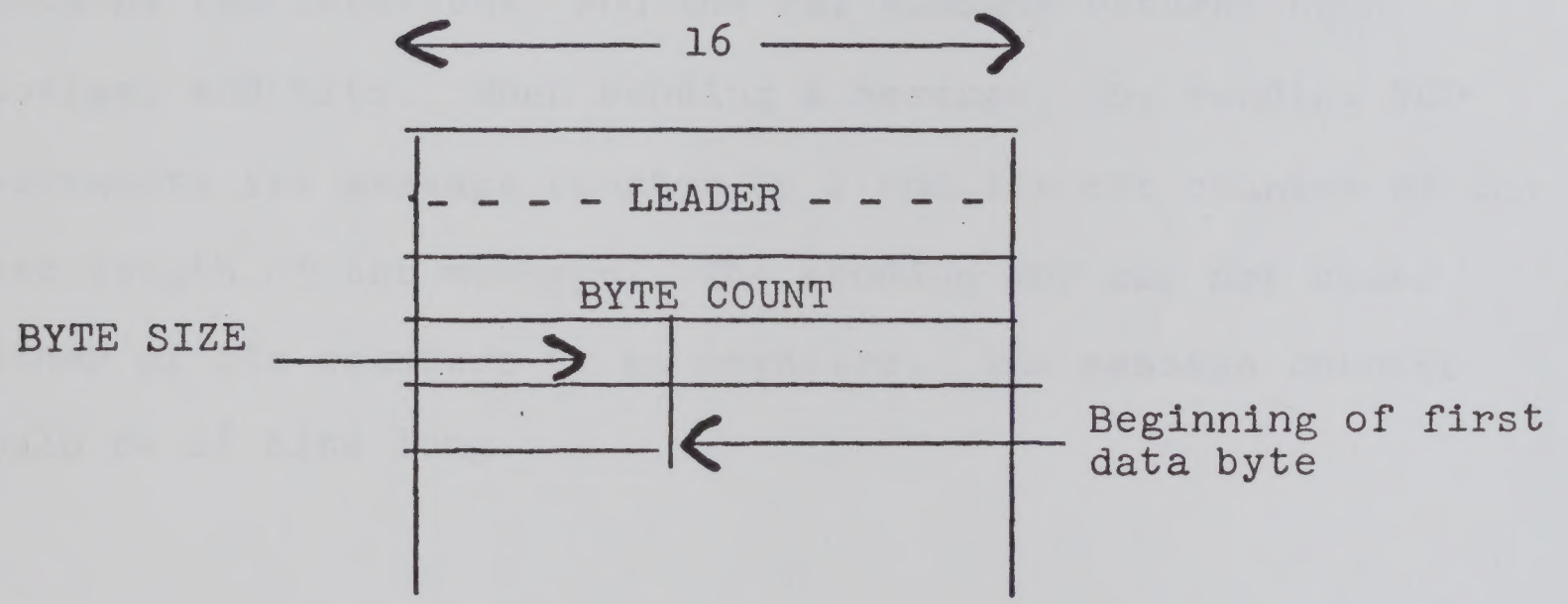
a) no special limit (∿ 1000 bytes)

b) 120 bytes

## 2. Message Format

It was agreed to abandon marking and include the text length in the form of a byte count and byte size. Unresolved was where to begin the first byte of data. The two choices are:

a) have the first data byte begin after 72 bits of leader, byte count, byte size and spacing. The message format would then be as in the diagram:




b) use the double physical transmission scheme presented in NWG/RFC 67. When sending a regular message, the Host would send a leader, byte count and byte size and terminate transmission. The second transmission would be the data.

At the receiving end, the IMP would transmit 64 bits of leader, byte count, byte size and spacing, and stop transmission. The next transmission would be only the data.

## 3. Allocation

With respect to the allocation mechanism embodied in the ALL, GVB and RET commands, two alternatives were proposed:

a) make no change.

b) The flow control algorithm should be changed to keep track of two quantities: messages and bits. The ALL, GVB, and RET commands each have two data fields. The ALL command allocates a message limit and a bit limit. Thw GVB command contains two fractions, and the RET command returns both messages and bits. When sending a message, the sending NCP decrements its message counter by 1 and its bit counter by the text length of the message. The sending NCP may not cause either of its counters to go negative. The message counter would be 16 bits long.